CONTRIBUTING TO THE OpenJDK

OR HOW TO ENGAGE IN THE OPENJDK PROJECT

Volker Simonis [Фолкер Симонис], SAP / volker.simonis@gmail.com

THE OPENJDK PROJECT

http://openjdk.java.net/

- Open Source implementation of Java SE
- Licensed under **GPLv2** (with Classpath exception)
- Reference implementation for Java 7, 8, 9, ..
- Collaboration space for different implementers:
  - RedHat, SAP, IBM, Google, Intel, Twitter, Azul
- Playground for Java SE related projects:
  - Coin, InvokeDynamic, Lambda, Nashorn, ...
  - Jigsaw, Graal, Panama, Value Types, ...

THE ORACLE CONTRIBUTION AGREEMENT (**OCA**)

- Copyright and patent sharing agreement with regard to your contribution
  - Simple dual licensing model
- Both, you and Oracle, can treat your contribution as if they were the sole owners
- Oracle guarantees to always publish any contribution under a suitable Free Software Foundation (FSF) or Open Source Initiative (OSI) approved license.

OPENJDK AND THE JAVA™

- You are not allowed to call it Java™ (or Java™ compatible) because:
  - "Java" is a registered trademark
  - You don't get access to the Technology Compatibility Kit (**TCK**)
- Accessing the TCK requires the signing of the OpenJDK Community TCK License Agreement (**OCTLA**)
  - Only for OpenJDK or "substantially derived" projects
  - OCTLA forbids disclosing TCK information to non-OCTLA licensees
- OCTLA is no trademark license!
- "OpenJDK" can be used according to the "OpenJDK Trademark Notice"

OPENJDK **BYLAWS**

- The "constitution" of the OpenJDK project
- Define OpenJDK **Groups**, **Projects** and **Roles**
- Define the **Governing Board** which manages the structure and operation of the OpenJDK community
  - Chair (appointed by Oracle): **Georges Saab**
  - Vice-Chair (appointed by IBM): **John Duimovich**
  - OpenJDK Lead (appointed by Oracle): **Mark Reinhold**
  - At-Large Member (elected by OpenJDK Members): **Doug Lea**, SUNY Oswego
  - At-Large Member (elected by OpenJDK Members): **Andrew Haley**, RedHat

OPENJDK ORGANISATION

- **Participants**: individuals who subscribed to an OpenJDK mailing list
- **Contributors**: Participants who signed the OCA
- **Groups** (currently 20)
  - consist of **Group Members** and a **Group Lead**
    - Group Members elect new Group Members from Contributors
  - sponsor Projects
- **Projects** (currently 54)
  - produce artifacts (code, documentation, whole JDK releases)
  - have code repositories, mailing lists and possibly web content
  - consist of **Authors**, **Committers**, **Reviewers** and a **Project Lead**
    - Project Leads appoint Authors from Contributors
    - Committers elect new Committers from Authors

OPENJDK INFRASTRUCTURE

- The OpenJDK Homepage
- A bunch of mailing lists (currently ~140)
- Too many Mercurial repositories
  - Every JDK is in fact a repository of 7 nested sub-repositories
- Code review server http://cr.openjdk.java.net
  - For review requests in Webrev format
  - Must be at least Author to get write access
- The JDK Bug System https://bugs.openjdk.java.net
- The OpenJDK Wiki https://wiki.openjdk.java.net

OPENJDK TOOLS - MERCURIAL

- Mercurial is a distributed source control system implemented in Python.
- OpenJDK uses nested sub-repositories so we must either:
  - clone them (corba, hotspot, jaxp, jaxws, jdk, langtools, nashorn) manually:

```
> hg clone http://hg.openjdk.java.net/jdk9/dev jdk9-dev
> cd jdk9-dev
> hg clone http://hg.openjdk.java.net/jdk8/jdk8/corba
...
> hg clone http://hg.openjdk.java.net/jdk8/jdk8/nashorn
```

  - or use the `get_source.sh` script which does the same in parallel:

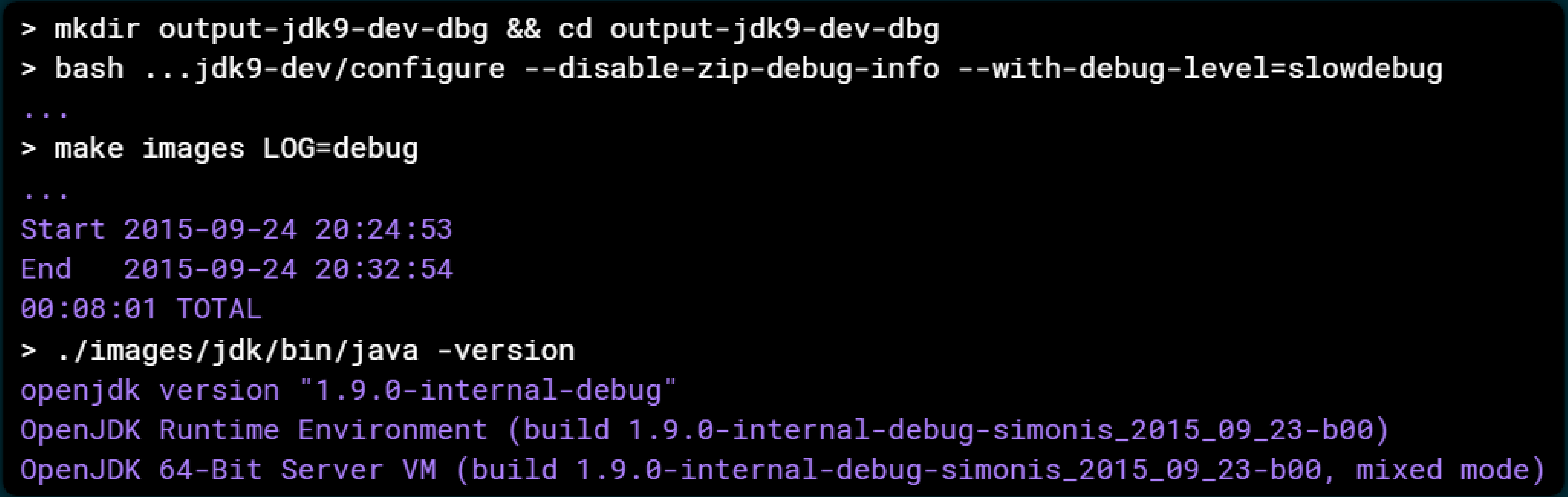
```
> hg clone http://hg.openjdk.java.net/jdk9/dev jdk9-dev
> cd jdk9-dev
> sh get_source.sh
```

  - Use `common/bin/hgforest.sh` to execute hg commands in all repositories

BUILDING THE OPENJDK

- Use a Supported Build Platform
- Favour Linux over Windows/MacOS X
- Use your favorite search engine extensively :)
- Have a look at the AdoptOpenJDK projekt
- Ask on build-dev@openjdk.java.net
- Unfortunately README-builds.html is terribbly outdated :(

BUILDING THE OPENJDK



```
> mkdir output-jdk9-dev-dbg && cd output-jdk9-dev-dbg
> bash ...jdk9-dev/configure --disable-zip-debug-info --with-debug-level=slowdebug
...
> make images LOG=debug
...
Start 2015-09-24 20:24:53
End   2015-09-24 20:32:54
00:08:01 TOTAL
> ./images/jdk/bin/java -version
openjdk version "1.9.0-internal-debug"
OpenJDK Runtime Environment (build 1.9.0-internal-debug-simonis_2015_09_23-b00)
OpenJDK 64-Bit Server VM (build 1.9.0-internal-debug-simonis_2015_09_23-b00, mixed mode)
```

On error use “`make images JOBS=1`” and take a look at “`build.log`”

OPENJDK TOOLS - JTreg

- JTreg is a Regression Test Harness for the OpenJDK platform
- The same harness which is used to run the TCK/JCK test suite
- Test can consist of single or multiple Java/C/C++/ShellScript/HTML files
  - Tests are automatically discovered by the harness trough embedded tags
  - Check the JTreg FAQ or have a look at the tests under jdk/test/
- Can execute JUnit and TestNG tests
- See JDK 9 Early Access Build Test Results and JDK 8 Update Releases Early Access Build Test Results
- Binaries are available from the AdoptOpenJDK project

OPENJDK TOOLS - WEBREV

- Webrev is a "poor man's" code review system (e.g. webrev for Bug 8081674)
- A collection of diffs and patches in different formats
- Ability to graphically display diffs in HTML-frames
- May include additional HTML-formatted documentation
- Ability to automatically link back to Java bugs at https://bugs.openjdk.java.net
- Simple Korn shell script available from http://hg.openjdk.java.net/code-tools/webrev

```
> cd jdk
> ksh webrev.ksh -o /tmp -O -c 8081674 -i 8081674.html
> unzip -l /tmp/webrev.zip
...
     5643  2015-09-25 17:36   webrev/raw_files/old/src/.../jni_util.c
     8643  2015-09-25 17:36   webrev/index.html
---------                     -------
   320695                     75 files
```

FINDING A BUG

- Search for the bug in the bug database https://bugs.openjdk.java.net
  - Not all bugs are visible (security bugs, customer bugs, ..)
- Try to reproduce the bug with the latest build (http://jdk8.java.net, http://jdk9.java.net)
- Try with a debug build (you'll have to build it!) - this may give you more insights
- Try to write a simple reproducer (if possible a `JTreg` test)

FIXING A BUG

- Bugs must be fixed in a development code line first!
- Fix the bug in the repository of the team that's responsible for the code
  - http://hg.openjdk.java.net/jdk9/client/jdk for AWT/Swing problems or http://hg.openjdk.java.net/jdk9/hs-comp/hotspot for a JIT bug, etc...
- Consider all the platforms the OpenJDK may run on
- Create a Webrev of your fix and upload it to your web-space
- Submit your fix to the appropriate mailing list:
  RFR(XS): 8132374: AIX: fix value of os.version property (Author request)
  [patch] JDK-4906983 (Contributor request)
- Be ready to discuss, defend and improve your fix several rounds

THE JAVA ENHANCEMENT PROCESS (**JEP**)

- Collect, review, sort, and record proposals for enhancements to the JDK
- JEP 1: JDK Enhancement-Proposal & Roadmap Process defines the process
- Simpler than a Java Community Process (**JCP**) Java Specification Request (**JSR**)
  - Specification changes still require a JSR
- The JEP process is still dominated by Oracle:
  - requires endorsement and funding by Group Leads
  - The OpenJDK Lead (appointed by Oracle) ultimately decides which JEPs to include into the Roadmap.
- Still attractive for “implementation enhancements” (i.e. new Ports, GCs, JITs, ..)